

DWES

Acceso a datos (PDO). CRUD

2º DAW

Trabajar con BBDD en PHP

❑ Ejercicio completo BBDD

Para realizar estos ejercicios utilizaremos la tabla *alumnos.sql* facilitada. Recuerda que puedes exportarla para tener una copia de seguridad en caso de que sea necesario.

El código puede estar en un solo script o separarlo en dos, uno para el formulario y otro para el proceso (a tu criterio).

Al pulsar el botón de envío del formulario se realizarán los siguientes pasos:

1. Se recogen los datos que nos han llegado del formulario.
2. Nos conectamos a la BD.
3. Hacer la consulta correspondiente utilizando una sentencia preparada. Los valores que se pasarán como parámetros son los elementos que nos han llegado del formulario.

Realizaremos las siguientes operaciones sobre BBDD:

- ✓ Insertar campos
- ✓ Buscar campos
- ✓ Borrar campos
- ✓ Modificar campos

Trabajar con BBDD en PHP

➤ Importar tabla desde PHPMyAdmin

1. **IMPORTANTE:** si vas a importar una tabla, la base de datos debe existir, la tabla depende de código sql, puede que la borre si existe o que de error.
2. Desde PHPMyAdmin importa el archivo **“alumnos.sql”** que está en el aula virtual del curso en la base de datos MyBDPDO.
3. Para poder hacerlo, tienes que tener seleccionada la base de datos en el panel de la izquierda.
4. A continuación pulsa el botón “importar” del menú superior.
5. En la pantalla que se mostrará, pulsa el botón “seleccionar archivo” y selecciona el archivo que te has descargado de la plataforma y comprueba que el conjunto de caracteres seleccionado está en UTF-8.
6. Pulsa el botón **“Continuar”**.
7. Comprueba los datos que se han cargado en la tabla.

Trabajar con BBDD en PHP

❑ Insertar campos

- Primero crearemos un formulario que reciba la información que queremos introducir en la BBDD
- En el formulario los elementos tienen el atributo "required". Esto nos asegura que no se va a enviar el formulario sin haber completado los campos (validación del cliente)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <style>
      form{
        width:30%;
      }
    </style>
  </head>
  <body>
    <h1>Alta de Alumnos</h1>
    <form action="" method="post">
      <fieldset>
        <legend>Datos del alumno</legend>
        Nombre del alumno:
        <input type="text" name="nombre" required><br><br>
        Apellidos del alumno:
        <input type="text" name="apellidos" required><br><br>
        Teléfono:
        <input type="text" name="telefono" required><br><br>
        mail de contacto:
        <input type="text" name="mail" required><br><br>
        <input type="submit" name="btnEnviar" value="Dar de alta">
      </fieldset>
    </form>
  </body>
</html>
```

Trabajar con BBDD en PHP

❑ Insertar campos

- En la parte del servidor, debemos comprobar si llegan datos del formulario.
- Si el formulario no se hubiera validado en el cliente, también deberíamos comprobar si el usuario introdujo todos los datos necesarios en él.
- Crearemos la consulta INSERT que permita introducir los datos introducidos
- Posteriormente ejecutaremos la consulta preparada haciendo uso de bindParam

```
<?php
$tblDatos = null;
$servidor = "localhost";
$usuario = "root";
$clave = "";
$sql="";

if(isset($_REQUEST['btnEnviar'])){
    try {
        $conn = new PDO("mysql:host=$servidor;dbname=myDBPDO;charset=utf8",
            $usuario, $clave);

        // asignamos el modo excepción
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $sql= "INSERT INTO ALUMNOS (nombre, apellidos, telefono, correo) "
            . "VALUES (:nom,:ape, :tel,:cor)";

        $stmt = $conn->prepare($sql);
        $stmt->bindParam(':nom', $_REQUEST['nombre']);
        $stmt->bindParam(':ape', $_REQUEST['apellidos']);
        $stmt->bindParam(':tel', $_REQUEST['telefono']);
        $stmt->bindParam(':cor', $_REQUEST['mail']);
        $stmt->execute();
        echo "Se ha insertado con éxito en myDBPDO <br/>";
    }
    catch(PDOException $e)
    {
        echo "Error: " . $e->getMessage();
    }
    $conn = null;
}

?>
```

Trabajar con BBDD en PHP

❑ Insertar campos

- Ojo, para evitar introducir duplicados, comprobamos si el alumno que queremos insertar ya existe en la tabla, ya que de ser así no deberíamos insertarlo.
- Para hacerlo, ejecutaremos (antes de insertar) una sentencia SELECT que cuente los registros que hay con el correo electrónico que nos llega desde el formulario, en principio la dirección de correo es única. Si hay algún registro con el mail que se introdujo en el formulario, significa que ese alumno ya existe en la base de datos y no debemos insertarlo.

```
<?php
$tblDatos = null;
$servidor = "localhost";
$usuario = "root";
$clave = "";
$sql="";
if(isset($_REQUEST['btnEnviar'])){
    try {
        $conn = new PDO("mysql:host=$servidor;dbname=myDBPDO;charset=utf8",
            $usuario, $clave);

        // asignamos el modo excepción
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        $sql= "SELECT COUNT(*) AS 'cantidad' FROM ALUMNOS WHERE correo='".$_REQUEST['mail']."'";
        $result=$conn->query($sql);
        $num=$result->fetch();
        if($num['cantidad']>0){
            echo "No se puede dar de alta, el alumno ya está matriculado en ese curso<br/>";
        }
        else{
            $sql= "INSERT INTO ALUMNOS (nombre, apellidos, telefono, correo) "
                . "VALUES (:nom,:ape, :tel,:cor)";

            $stmt = $conn->prepare($sql);
            $stmt->bindParam(':nom', $_REQUEST['nombre']);
            $stmt->bindParam(':ape', $_REQUEST['apellidos']);
            $stmt->bindParam(':tel', $_REQUEST['telefono']);
            $stmt->bindParam(':cor', $_REQUEST['mail']);
            $stmt->execute();
            echo "se ha insertado con éxito en myDBPDO <br/>";
        }
    }
    catch(PDOException $e)
    {
        echo "Error: " . $e->getMessage();
    }
    $conn = null;
}
```

Trabajar con BBDD en PHP

❑ Buscar campos

- Crearemos un formulario que nos permita incluir el campo de búsqueda sobre nuestra tabla en la BBDD.
- Generalmente suele ser un campo descriptivo, fácil de recordar o la clave primaria de nuestra tabla.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <style>
      form{
        width:30%;
      }
    </style>
  </head>
  <body>
    <h1>Buscar alumnos</h1>
    <p style="color:red;"><?=$msg?></p>
    <form action="" method="post">
      <fieldset>
        <legend>Datos del alumno a buscar</legend>
        Nombre del alumno:
        <input type="text" name="nombre" required><br><br/>
        <input type="submit" name="btnEnviar" value="Buscar">
      </fieldset>
    </form>
  </body>
</html>
```


Trabajar con BBDD en PHP

❑ Buscar campos

- Esta solución busca en la tabla de la base de datos los alum@s cuyo nombre coincida con el introducido en el formulario.
- Sin embargo, esta solución no es válida si el nombre a buscar forma parte de un nombre. Por ejemplo todas las alumnas que en su nombre lleven la palabra María.
- Utilizando ->fetchAll() recupero de una sola vez todos los registros, o un array vacío si no encuentra ninguno.

```
<?php
$msg="";
if(isset($_REQUEST['btnEnviar'])) {
    $servidor = "localhost";
    $usuario = "root";
    $clave = "";

    // Comprobamos si nos ha llegado el dato a buscar
    try {
        $conn = new PDO("mysql:host=$servidor;dbname=myDBPDO;charset=utf8",
            $usuario, $clave);
        // asignamos el modo excepción
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $sql = "SELECT * FROM ALUMNOS WHERE NOMBRE=:nom;";
        $stmt = $conn->prepare($sql);
        $stmt->bindParam(':nom', $_REQUEST['nombre']);
        $stmt->execute();
        $registros = $stmt->fetchAll(PDO::FETCH_ASSOC);
        //Comprobamos si se han encontrado registros
        if(count($registros)>0) {
            echo "<h2>Alumn@s encontrados que contienen en el nombre".$_REQUEST['nombre']."</h2>";
            echo "<ul>";
            foreach($registros as $fila) {
                echo "<li>código: ".$fila["CODIGO"]. ", Nombre: ".$fila["NOMBRE"].
                    " ". $fila["APELLIDOS"]. ", Teléfono: ".$fila["TELEFONO"].
                    " , correo electrónico: ".$fila["CORREO"]. "</li>";
            }
            echo "</ul>";
        }
        else {
            $msg="No se ha encontrado ningún alumn@ que contenga".$_REQUEST['nombre']." en su nombre";
        }
    }
    catch(PDOException $e)
    {
        echo "Error: " . $e->getMessage();
    }
    $conn = null;
}
?>
```


Trabajar con BBDD en PHP

❑ Buscar campos

- Esta solución busca en la base de datos las personas que contengan en su nombre la palabra que se introdujo en el formulario utilizando en la **sentencia SQL “LIKE”**.
- El marcador de posición “%” se añade en el parámetro, si no da error. Además lo separe en una variable donde lo uno con el nombre recibido del formulario porque si no da error.

```
<?php
$msg="";
if(isset($_REQUEST['btnEnviar'])){
    $servidor = "localhost";
    $usuario = "root";
    $clave = "";

    // Comprobamos si nos ha llegado el dato a buscar
    try {
        $conn = new PDO("mysql:host=$servidor;dbname=myDBPDO;charset=utf8",
            $usuario, $clave);
        // asignamos el modo excepción
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $sql= "SELECT * FROM ALUMNOS WHERE NOMBRE LIKE :nom;";

        $stmt = $conn->prepare($sql);
        $nombre="%. $_REQUEST['nombre'] . %";
        $stmt->bindParam(':nom', $nombre);
        $stmt->execute();

        $registros = $stmt->fetchAll(PDO::FETCH_ASSOC);
        //Comprobamos si se han encontrado registros
        if(count($registros)>0){
            echo "<h2>Alumn@s encontrados que contienen en el nombre". $_REQUEST['nombre'] . "</h2>";
            echo "<ul>";
            foreach($registros as $fila){
                echo "<li>código: ".$fila["CODIGO"]. " , Nombre: ".$fila["NOMBRE"].
                    " ". $fila["APELLIDOS"]. " , Teléfono: ".$fila["TELEFONO"].
                    " , correo electrónico: ".$fila["CORREO"] . "</li>";
            }
            echo "</ul>";
        }
        else
        {
            $msg="No se ha encontrado ningún alumn@ que contenga". $_REQUEST['nombre'] . " en su nombre";
        }
    }
    catch(PDOException $e)
    {
        echo "Error: " . $e->getMessage();
    }
    $conn = null;
}
?>
```

Trabajar con BBDD en PHP

❑ Borrar campos

- Crearemos un formulario que nos permita incluir el campo a borrar sobre nuestra tabla en la BBDD.
- Generalmente suele ser un campo descriptivo, fácil de recordar o la clave primaria de nuestra tabla.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <style>
      form{
        width:30%;
      }
    </style>
  </head>
  <body>
    <h1>Borrar alumno</h1>
    <p style="color:red;"><?=$msg?><p>
    <form action="" method="post">
      <fieldset>
        <legend>Datos del alumno a borrar</legend>
        Código del alumno:
        <input type="text" name="codigo" required><br><br/>
        <input type="submit" name="btnEnviar" value="Borrar">
      </fieldset>
    </form>

  </body>
</html>
```

Trabajar con BBDD en PHP

❑ Borrar campos

- Se comprueba si el alumn@ está en la base de datos, en cuyo caso se procede a la eliminación de sus datos.
- En caso de no existir, se mostrará un mensaje indicándolo.
- Utilizamos la función intval() antes de almacenar el valor en la BBDD para asegurar que se guarda con el tipo de dato definido en el esquema.

```
<?php
$msg="";
if(isset($_REQUEST['codigo'])){
    $servidor = "localhost";
    $usuario = "root";
    $clave = "";

    try {
        $conn = new PDO("mysql:host=$servidor;dbname=myDBPDO;charset=utf8",
            $usuario, $clave);

        // asignamos el modo excepción
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        //Comprobamos que el registro a borrar existe en la BD
        $sql = "SELECT CODIGO FROM ALUMNOS WHERE codigo=:cod;";
        $stmt = $conn->prepare($sql);
        $stmt->bindParam(':cod',intval($_REQUEST['codigo']));
        $stmt->execute();
        $registros = $stmt->fetch(PDO::FETCH_ASSOC);
        if($registros>0){
            $sql = "DELETE FROM ALUMNOS WHERE codigo=:cod;";
            $stmt = $conn->prepare($sql);
            $stmt->bindParam(':cod',intval($_REQUEST['codigo']));
            $stmt->execute();
            $msg= "El alumn@ se ha eliminado correctamente";
        }
        else{
            $msg= "El alumn@ a borrar NO existe en la BD";
        }
    }
    catch(PDOException $e)
    {
        echo "Error: " . $e->getMessage();
    }
    $conn = null;
}
else{
    $msg="Hay que introducir un código a borrar";
}
```

Trabajar con BBDD en PHP

❑ Modificar campos

En función del valor de la variable \$datosCorrectos va a mostrar el formulario de introducción del código del usuario a modificar, o el formulario con los datos de el alumno indicado que hay actualmente en la base de datos.

```
<body>
<h1>Modificar alumno</h1>
<p style="color:red;"><?=$msg?></p>
<?php
    $html="";
    if(!$datosCorrectos){
        $html="<form action='' method='post'>";
        $html.="<fieldset><legend>Código del alumno a modificar</legend>";
        $html.="Código del alumno: <input type='text' name='codigo' required><br><br/>";
        $html.="<input type='submit' name='btnEnviar' value='Buscar'>";
        $html.="</fieldset></form>";
    }
    else{
        $html="<form action='update_process.php' method='post'>";
        $html.="<fieldset><legend>Datos actuales del alumno a modificar</legend>";
        $html.="Nombre: <input type='text' name='nombre' value='".$reg['NOMBRE']."'><br><br/>";
        $html.="Apellidos: <input type='text' name='apellidos' value='".$reg['APELLIDOS']."'><br><br/>";
        $html.="Teléfono: <input type='text' name='telefono' value='".$reg['TELEFONO']."'><br><br/>";
        $html.="Correo electrónico: <input type='text' name='correo' value='".$reg['CORREO']."'><br><br/>";
        $html.="<input type='hidden' name='hiddenCodigo' value='".$reg['CODIGO']."'><br><br/>";
        $html.="<input type='submit' name='btnModificar' value='Modificar'>";
        $html.="</fieldset></form>";
        $html.="";
    }
    echo $html;
?>
</body>
```

En el código un párrafo que muestra el contenido de una variable \$msg en color rojo, que utilizaremos para mostrar mensajes de error. Debe inicializarse en la parte de PHP como cadena vacía, si no dará error al intentar mostrar una variable que no existe.

Trabajar con BBDD en PHP

❑ Modificar campos

En la parte de código PHP comprobamos si nos ha llegado algún código (se ha pulsado el botón del primer formulario). Si es así, busca en la base de datos los datos del alumno, si lo encuentra cambia el valor de la variable \$datosCorrectos a true para que pinte el formulario con los datos encontrados en la base de datos y en el que puede modificarlos.

```
<?php
/* comprobamos si ha escrito algo en el campo del formulario de la página */
$msg="";
$datosCorrectos=false;
if(isset($_REQUEST['codigo'])) {
    $servidor = "localhost";
    $usuario = "root";
    $clave = "";

    try {
        $conn = new PDO("mysql:host=$servidor;dbname=myDBPDO;charset=utf8",
            $usuario, $clave);
        // asignamos el modo excepción
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        //buscamos si existe el alumno a modificar
        $sql = "SELECT * FROM ALUMNOS WHERE codigo=:cod;";
        $stmt = $conn->prepare($sql);
        $stmt->bindParam(':cod',intval($_REQUEST['codigo']));
        $stmt->execute();
        $reg = $stmt->fetch(PDO::FETCH_ASSOC);
        //si lo encuentra pinta un formulario para introducir los nuevos datos
        if($reg<=0){
            $msg= "El alumno a modificar NO existe en la BD";
            $datosCorrectos=false;
        }
        else
            $datosCorrectos=true;
    }
    catch(PDOException $e)
    {
        echo "Error: " . $e->getMessage();
    }
    $conn = null;
}
```

Trabajar con BBDD en PHP

❑ Modificar campos

En la parte de código PHP comprobamos si se pulsó el botón modificar del formulario de la página index_update.php en la que se cambiaban los datos del alumno. Si es así, prepara y ejecuta la sentencia SQL que actualiza los datos en la base de datos.

Este código podría mejorarse si antes de actualizar la base de datos se comprueba si se ha cambiado algún dato con respecto los datos que había en la base de datos, y en caso de que no se haya modificado nada, no hacer la actualización.

```
<?php
$msg="";
if(isset($_REQUEST['btnModificar'])){
    $servidor = "localhost";
    $usuario = "root";
    $clave = "";

    try {
        $conn = new PDO("mysql:host=$servidor;dbname=myDBPDO;charset=utf8",
            $usuario, $clave);
        // asignamos el modo excepción
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        //buscamos si existe el alumno a modificar
        $sql = "UPDATE ALUMNOS SET NOMBRE=:nom, APELLIDOS=:ape, "
            . "TELEFONO=:tel, CORREO=:correo WHERE codigo=:cod;";
        $stmt = $conn->prepare($sql);
        $stmt->bindParam(':nom', $_REQUEST['nombre']);
        $stmt->bindParam(':ape', $_REQUEST['apellidos']);
        $stmt->bindParam(':tel', $_REQUEST['telefono']);
        $stmt->bindParam(':correo', $_REQUEST['correo']);
        $stmt->bindParam(':cod', intval($_REQUEST['codigo']));
        if($stmt->execute())
            $msg="El alumno se ha actualizado correctamente";
    }
    catch(PDOException $e)
    {
        echo "Error: " . $e->getMessage();
    }
    $conn = null;
}
else{
    $msg="Hay algún error en los datos a modificar, vuelva a la página principal";
}
?>
```


Trabajar con BBDD en PHP

❑ Modificar campos

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
    <p style="color:red"><?php echo $msg?></p>
    <a href='index_update.php'>Volver a la página principal</a>
  </body>
</html>
```

En la parte de código HTML utilizamos un párrafo para mostrar un mensaje. Debe estar definido en la parte de PHP con una cadena vacía para que no de error al intentar mostrarlo.

Además tenemos un enlace que lleva a la página principal por si se desea modificar más de un alumno.